

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-039186

(43)Date of publication of application : 12.02.1999

(51)Int.Cl.

G06F 11/22
G06F 9/445
G06F 11/28

(21)Application number : 09-192657

(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD

(22)Date of filing : 17.07.1997

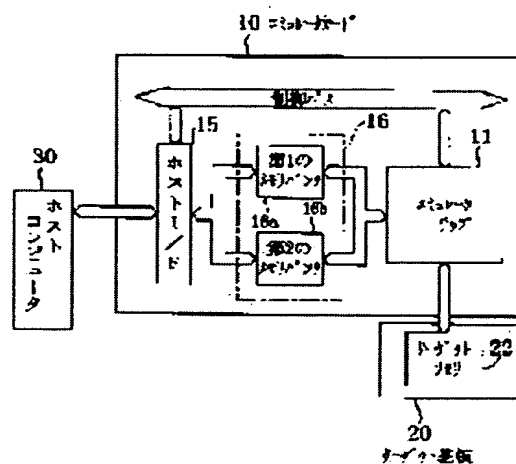
(72)Inventor : NAGATA SHOJI

(54) DEBUG SUPPORTING DEVICE AND DOWN LOADING METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To reduce a down loading time of a user program to a target substrate in the debugging of a user program by using a debug supporting device having an emulation function.

SOLUTION: An emulator card 10 which operates as a debug supporting device is provided with a buffer memory 16 being a buffer for data transfer between a host computer 30 and a target substrate 20. The buffer memory 16 has two memory banks 16a and 16b which parallelly execute access from the computer 30 to one direction and access from an emulation chip 11 to the other. At the time of down loading, transfer from the computer 30 to the memory 16 and transfer from the memory 16 to the substrate 20 are parallelly and repeatedly performed. This eliminates waiting state of the computer 30 and the chip 11 at the time of down loading and realizes down loading which is faster than in the conventional practice.



LEGAL STATUS

[Date of request for examination] 31.03.2000

[Date of sending the examiner's decision of rejection] 27.11.2001

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-39186

(43) 公開日 平成11年(1999) 2月12日

(51) Int.Cl. ⁶	識別記号	F I
G 0 6 F 11/22	3 4 0	G 0 6 F 11/22 3 4 0 A
9/445		11/28 3 4 0 A
11/28	3 4 0	9/06 4 2 0 H

審査請求 未請求 請求項の数4 O L (全 6 頁)

(21) 出願番号 特願平9-192657

(22) 出願日 平成9年(1997) 7月17日

(71) 出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72) 発明者 永田 昭二

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

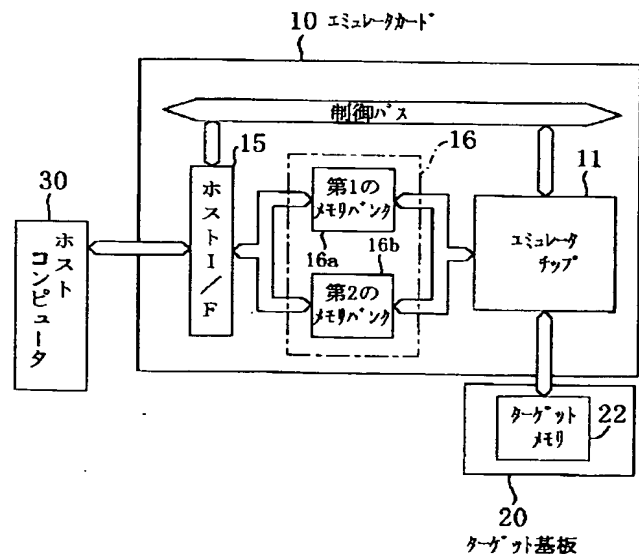
(74) 代理人 弁理士 前田 弘 (外2名)

(54) 【発明の名称】 デバッグ支援装置およびダウンロード方法

(57) 【要約】

【課題】 エミュレーション機能を有するデバッグ支援装置を用いたユーザプログラムのデバッグにおいて、ターゲット基板へのユーザプログラムのダウンロード時間を短縮する

【解決手段】 デバッグ支援装置としてのエミュレータカード10は、ホストコンピュータ30とターゲット基板20との間のデータ転送のバッファとなるバッファメモリ16を備えている。バッファメモリ16は、一方へのホストコンピュータ30からのアクセスと他方へのエミュレーションチップ11からのアクセスとが並列に実行可能な2つのメモリバンク16a、16bを備えている。ダウンロードの際には、ホストコンピュータ30からバッファメモリ16への転送とバッファメモリ16からターゲット基板20への転送とを並列に繰り返し行う。これにより、ダウンロードの際にホストコンピュータ30やエミュレータチップ11の待ち状態がなくなり、従来よりも高速なダウンロードを実現することができる。



【特許請求の範囲】

【請求項1】 ターゲット基板と接続され、このターゲット基板における被デバッグプログラムの動作をホストコンピュータの制御によりエミュレーションし、被デバッグプログラムのデバッグを支援するデバッグ支援装置であって、

ターゲット基板に搭載されるマイコンの代わりに被デバッグプログラムを実行するエミュレーションチップと、ホストコンピュータとターゲット基板との間のデータ転送のバッファとなるバッファメモリとを備え、前記バッファメモリは、ホストコンピュータからのアクセスとエミュレーションチップからのアクセスとを並列に実行可能に構成されていることを特徴とするデバッグ支援装置。

【請求項2】 請求項1記載のデバッグ支援装置において、

前記バッファメモリは、複数のメモリバンクによって構成されており、一のメモリバンクへのホストコンピュータからのアクセスと、他のメモリバンクへのエミュレーションチップからのアクセスとが並列に実行可能であることを特徴とするデバッグ支援装置。

【請求項3】 請求項1記載のデバッグ支援装置において、

前記バッファメモリは、単一のメモリバンクを領域分割して構成されており、一のメモリ領域へのホストコンピュータからのアクセスと、他のメモリ領域へのエミュレーションチップからのアクセスとが並列に実行可能であることを特徴とするデバッグ支援装置。

【請求項4】 ターゲット基板における被デバッグプログラムの動作をホストコンピュータの制御によりエミュレーションするデバッグ支援装置を用いて被デバッグプログラムのデバッグを行う際に、被デバッグプログラムをターゲット基板に転送するダウンロード方法であって、

前記デバッグ支援装置は、ターゲット基板に搭載されるマイコンの代わりに被デバッグプログラムを実行するエミュレーションチップと、ホストコンピュータからのアクセスとエミュレーションチップからのアクセスとを並列に実行可能に構成されたバッファメモリとを有し、かつ、被デバッグプログラムの一部を、ホストコンピュータから前記バッファメモリに転送する工程と、前記バッファメモリに格納された被デバッグプログラムの一部を、エミュレーションチップの制御によりターゲット基板に転送する工程とを並列に繰り返し行いながら、被デバッグプログラムを転送することを特徴とするダウンロード方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、ターゲット基板に

おけるユーザープログラムの動作をエミュレートし、ユーザープログラムのデバッグを支援するデバッグ支援装置に関するものであり、特に、ユーザープログラムのターゲット基板へのダウンロード時間を短縮する技術に関する。

【0002】

【従来の技術】 従来のデバッグ支援装置において、特にインサーキットエミュレータ（以後エミュレータと称す）と呼ばれる装置は、少なくとも、被デバッグプログラム（ユーザープログラム）のロードから、メモリやCPUレジスタの読み書き、プログラム実行及び停止、実行時間測定や実行軌跡のトレース採取、イベント監視といったプログラムデバッグのため必要な機能を備えており、マイコンのアプリケーション開発において広く用いられる装置である。

【0003】

【発明が解決しようとする課題】 従来のエミュレータでは、ホストコンピュータからターゲット基板へのユーザープログラムのダウンロードに時間がかかりすぎるという問題があった。

【0004】 従来のユーザープログラムのダウンロードは、単純化して説明すると、まず、ホストコンピュータの例えばディスク装置内に格納されたユーザープログラムのコードをエミュレータのモニタメモリに読み込み、その後、エミュレータのエミュレータチップがモニタプログラムに従って、モニタメモリに格納されたユーザープログラムをターゲット基板に転送するという手順を採っていた。実際には、ユーザープログラムのサイズはモニタメモリのバッファサイズよりも大きいことが多いので、ユーザープログラムはモニタメモリのバッファサイズを単位として分割され、前記のような手順の繰り返しによって、ユーザープログラムの全てのコードをターゲット基板に転送する。

【0005】 このとき、ホスト・ターゲット間のハンドシェイク制御すなわちバッファサイズ単位の転送の順序性確保を行うべく、ホストコンピュータがモニタメモリへユーザープログラムの書き込みを行う間はエミュレーションチップはアイドルループを行いながらその書き込みの完了を待つ一方、エミュレーションチップがモニタメモリからユーザープログラムの読み出しを行う間はホストコンピュータはアイドルループを行いながらその読み出しの完了を待つ。

【0006】 しかしながら、ホストコンピュータおよびエミュレーションチップが互いの動作をアイドルループしながら待つため、ダウンロードにおけるホストコンピュータおよびエミュレーションチップの動作効率が悪く、結果として長いダウンロード時間を要していた。

【0007】 このダウンロード時間の問題は、近年、デバッグ対象のアプリケーションプログラムの規模の増大傾向、ならびに開発期間の短縮傾向に伴い大きな問題と

なっており、デバッグ支援システムにおいてユーザープログラムのダウンロード時間を短縮することは、重要な課題となっている。

【0008】前記のような問題に鑑み、本発明は、ターゲット基板におけるユーザープログラムの動作をエミュレーションするデバッグ支援装置を用いたユーザープログラムのデバッグにおいて、ターゲット基板へのユーザープログラムのダウンロード時間を短縮することを目的とする。

【0009】

【課題を解決するための手段】前記の課題を解決するため、請求項1の発明が講じた手段は、ターゲット基板と接続され、このターゲット基板における被デバッグプログラムの動作をホストコンピュータの制御によりエミュレーションし、被デバッグプログラムのデバッグを支援するデバッグ支援装置として、ターゲット基板に搭載されるCPUの代わりに被デバッグプログラムを実行するエミュレーションチップと、ホストコンピュータとターゲット基板との間のデータ転送のバッファとなるバッファメモリとを備え、前記バッファメモリは、ホストコンピュータからのアクセスとエミュレーションチップからのアクセスとを並列に実行可能に構成されているものである。

【0010】請求項1の発明によると、バッファメモリは、ホストコンピュータからのアクセスとエミュレーションチップからのアクセスとを並列に実行可能に構成されているので、被デバッグプログラムのダウンロードなどのホストコンピュータとターゲット基板との間のデータ転送において、ホストコンピュータとバッファメモリとの間の転送と、エミュレーションチップによって制御されるバッファメモリとターゲット基板との間の転送とが、並列に実行することができる。このため、ホストコンピュータおよびエミュレーションチップは、データ転送過程において実行待ちをする必要がなくなり、これにより、ダウンロード時間を大幅に短縮することができる。

【0011】そして、請求項2の発明では、前記請求項1のデバッグ支援装置におけるバッファメモリは、複数のメモリバンクによって構成されており、一のメモリバンクへのホストコンピュータからのアクセスと他のメモリバンクへのエミュレーションチップからのアクセスとが並列に実行可能であるものとする。

【0012】また、請求項3の発明では、前記請求項1のデバッグ支援装置におけるバッファメモリは、単一のメモリバンクを領域分割して構成されており、一のメモリ領域へのホストコンピュータからのアクセスと、他のメモリ領域へのエミュレーションチップからのアクセスとが並列に実行可能であるものとする。

【0013】また、請求項4の発明が講じた解決手段は、ターゲット基板における被デバッグプログラムの動

作をホストコンピュータの制御によりエミュレーションするデバッグ支援装置を用いて被デバッグプログラムのデバッグを行う際に、被デバッグプログラムをターゲット基板に転送するダウンロード方法として、前記デバッグ支援装置は、ターゲット基板に搭載されるCPUの代わりに被デバッグプログラムを実行するエミュレーションチップと、ホストコンピュータからのアクセスとエミュレーションチップからのアクセスとを並列に実行可能に構成されたバッファメモリとを有し、かつ、被デバッグプログラムの一部を、ホストコンピュータから前記バッファメモリに転送する工程と、前記バッファメモリに格納された被デバッグプログラムの一部を、エミュレーションチップの制御によりターゲット基板に転送する工程とを並列に繰り返し行いながら、被デバッグプログラムを転送するものである。

【0014】請求項4の発明によると、ホストコンピュータおよびエミュレーションチップが実行待ちをすることなく、被デバッグプログラムのダウンロードを行うことができ、これにより、ダウンロード時間を大幅に短縮することができる。

【0015】

【発明の実施の形態】以下、本発明の実施の形態について、図面を参照して説明する。

【0016】図1は本発明の一実施形態に係るデバッグ支援装置を含む全体構成を示す図である。図1において、10はデバッグ支援装置としてのエミュレータカード、20はデバッグの対象となるユーザープログラム（被デバッグプログラム）が動作するターゲット基板、30はエミュレータカード10のエミュレーション動作を制御するホストコンピュータである。エミュレータカード10はエミュレーションチップ11、モニタメモリ12、エミュレーションメモリ13およびトレースメモリ14を備えており、ターゲット基板20はマイコン21、ターゲットメモリ22およびユーザ回路23を備えている。エミュレータカード10はターゲット基板20と接続されて用いられ、デバッグの際にはエミュレーションチップ11がターゲット基板20のマイコン21の代わりにユーザープログラムを実行する。モニタメモリ12は、エミュレータチップ11によって実行されるモニタプログラム及びこのモニタプログラムの実行に用いるデータを格納する。エミュレーションメモリ13は、ターゲット基板20が未完成のときでもアプリケーションのデバッグが可能になるよう、ターゲットメモリ22を代替する。

【0017】モニタプログラムは、エミュレータチップ11のレジスタやメモリの読み書きやユーザープログラムの実行及び停止などのデバッグ機能の最もプリミティブな機能を備え、ホストコンピュータ上のデバッグソフトウェアと連係して動作するものであり、デバッグソフトウェアはモニタプログラムとの通信機能、およびエミ

ュレータカード10が備えるその他の機能（実行時間測定や実行軌跡のトレース、イベント監視機能等）を用いることにより高水準の（例えばシンボリックデバッグ機能やCソースレベルデバッグ機能、プロファイル機能等）デバッグ機能を提供するものである。

【0018】図2は本発明の一実施形態に係るデバッグ支援装置において、ユーザープログラムのダウンロード動作に係わる構成の概略を示すブロック図である。ここで、モニタメモリ12はモニタプログラムが格納されると共にホストコンピュータ上のデバッグソフトウェアと通信を行うため必要なバッファ領域を備えており、いわゆるバッファメモリとしての機能を有している。図2において、モニタメモリ12は、バッファメモリ16として、第1のメモリバンク16aおよび第2のメモリバンク16bを備えている。エミュレータカード10は、ホストインターフェース11を介して接続されたホストコンピュータ30で動作するデバッグソフトウェアによるリード及びライト動作に従って、全体が制御される。

【0019】ホストコンピュータ30からターゲットメモリ22に被デバッグプログラムとしてのユーザープログラムのダウンロードを行う際には、デバッグソフトウェア及びモニタプログラムは、相互に定められるメモリ書き込みの通信プロトコルによって通信し、モニタメモリ12中のバッファメモリ16を経由させる処理を備えることにより、ターゲットメモリ22への書き込みを行なう。

【0020】図2に示す本発明の実施形態におけるユーザープログラムのダウンロード方法について、図3を用いて説明する。

【0021】まず、図3に示すように、ホストコンピュータ30が、デバッグソフトウェアに従い、ホストコンピュータ30の例えばディスク装置内に格納されているユーザープログラムのコードを第1のメモリバンク16aに書き込む。次に、エミュレーションチップ11がモニタプログラムに従い、第1のメモリバンク16aに格納されたユーザープログラムをターゲットメモリ22に転送すると共に、その転送の完了を待つ間に、ホストコンピュータ30が第2のメモリバンク16bにユーザープログラムの後続のコードを格納する。

【0022】さらに、第2のメモリバンク16bに格納されたユーザープログラムをエミュレーションチップ11がターゲットメモリ22へ転送すると共に、その転送の完了を待つ間に、ホストコンピュータ30が第1のメモリバンク16aにユーザープログラムの後続のコードを格納する。このように、第1および第2のメモリバンク16a、16bは、一方へのホストコンピュータ30からのアクセスと他方へのエミュレーションチップ11からのアクセスとが並列に実行可能であり、ホストコンピュータ30からバッファメモリ16へのユーザープログラムの書き込みと、バッファメモリ16に格納された

ユーザープログラムのターゲットメモリ22への読み出しとが並列に行われる。

【0023】実際には、ユーザープログラムのサイズは、各メモリバンク16a、16bのサイズよりも大きいので、ユーザープログラムをメモリバンクのサイズを単位に分割され、前記のような手順の繰り返しによって、コード全体がターゲットメモリ22に転送される。

【0024】言い換えると、本実施形態に係るダウンロード方法は、図4に示す従来のダウンロード方法と比較すると分かるように、ホストコンピュータ30がバッファメモリ16への書き込みを行う間は、その書き込み完了までの待ち時間を利用して、エミュレータチップ11はそれ以前に格納されたユーザープログラムのコードをバッファメモリ16からターゲットメモリ22に転送し、エミュレータチップ11がバッファメモリ16からターゲットメモリ22への書き込みを行う間は、その書き込み完了までの待ち時間を利用して、ホストコンピュータ30は次のユーザープログラムのコードをバッファメモリ16に格納するものである。

【0025】以上説明したようなエミュレータカード10の構成とダウンロード方法を採用することによって、ユーザープログラムのダウンロード動作において、ホストコンピュータ30およびエミュレーションチップ11のアイドル時間を減少させることができ、ダウンロード動作をパイプライン的に処理させることができるので、ユーザープログラムのダウンロードに要する時間を従来よりも短縮することができる。

【0026】なお、図2では、バッファメモリ16は2つのメモリバンクによって構成されているものとしたが、3以上のメモリバンクからなる構成としても良い。

【0027】図5は本実施形態に係るデバッグ支援装置の構成の変形例を示すブロック図である。図5において、図2と異なるのは、モニタメモリ12が、第1および第2のメモリバンク16a、16bからなるバッファメモリ16の代わりに、単一のメモリバンクからなるバッファメモリ17を備えている点である。バッファメモリ17は第1のメモリ領域17aと第2のメモリ領域17bとに領域分割されており、第1および第2のメモリ領域17a、17bは、一方へのホストコンピュータ30からのアクセスと他方へのエミュレーションチップ11からのアクセスとが並列に実行可能に構成されている。

【0028】なお、図5では、バッファメモリ17は2つのメモリ領域に分割されたものとしたが、3以上のメモリ領域に分割しても良い。

【0029】なお、本実施形態では、ユーザープログラムをターゲット基板20のターゲットメモリ22にダウンロードするものとしたが、ターゲット基板20が未完成のときにアプリケーションのデバッグを行うべく、ターゲットメモリ22の代わりにエミュレーションメモリ

13にユーザープログラムをダウンロードする場合でも、本実施形態と同様に実施可能である。

【0030】

【発明の効果】以上説明したように、本発明によると、被デバッグプログラムをバッファメモリを経由してターゲット基板に高速にダウンロードすることができ、その効果は大きい。

【図面の簡単な説明】

【図1】本発明の一実施形態に係るデバッグ支援装置を含む、一般的なデバッグ支援システムの全体構成図である。

【図2】本発明の一実施形態に係るデバッグ支援装置の構成を示すブロック図である。

【図3】本発明の一実施形態に係るデバッグ支援装置におけるダウンロードの動作を示すタイミングチャートである。

【図4】従来のデバッグ支援装置におけるダウンロード

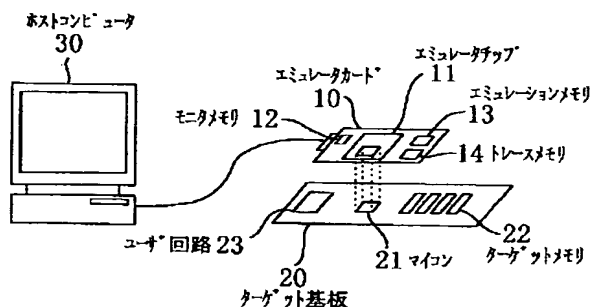
の動作を示すタイミングチャートである。

【図5】本発明の一実施形態に係るデバッグ支援装置の他の構成例を示すブロック図である。

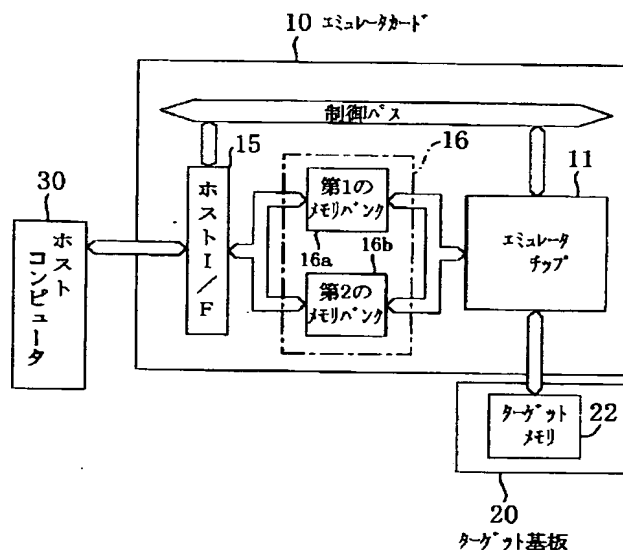
【符号の説明】

- 10 エミュレータカード（デバッグ支援装置）
- 11 エミュレータチップ
- 12 モニタメモリ
- 16 バッファメモリ
- 16a 第1のメモリバンク
- 16b 第2のメモリバンク
- 17 バッファメモリ
- 17a 第1のメモリ領域
- 17b 第2のメモリ領域
- 20 ターゲット基板
- 21 マイコン
- 30 ホストコンピュータ

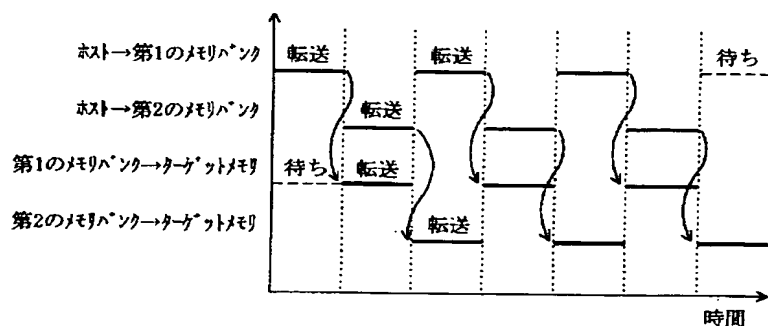
【図1】



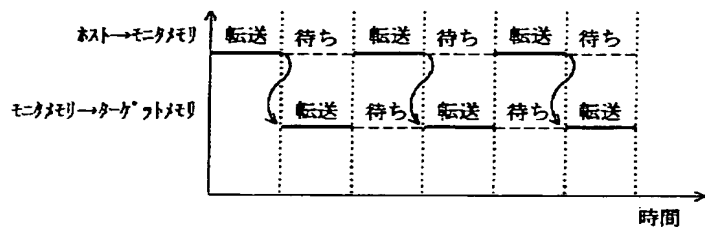
【図2】



【図3】



【図4】



【図5】

